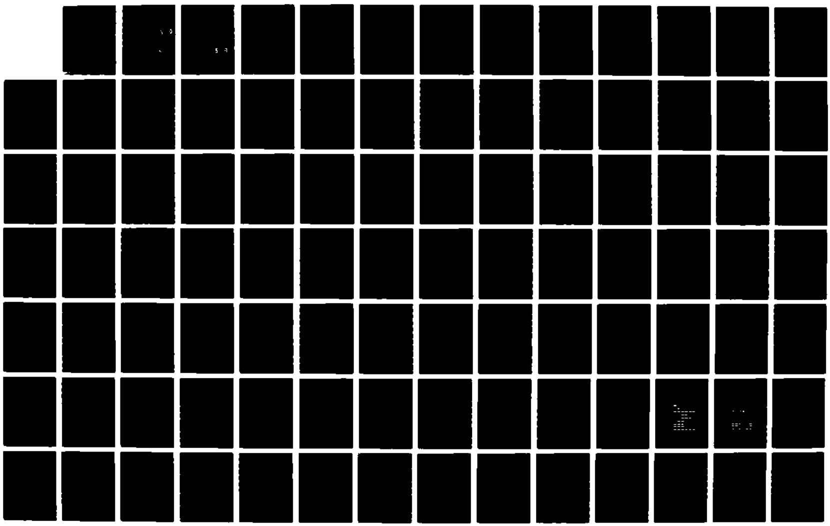
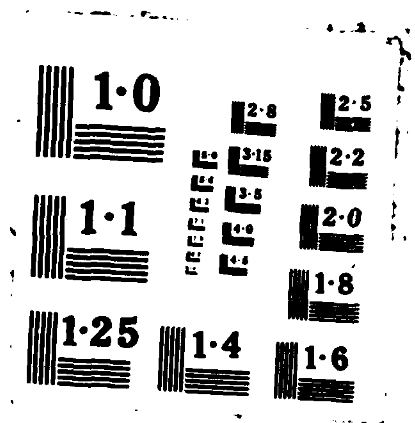
AD-A188 957 A COMPARISON OF INFORMATION SYSTEM DEVELOPMENT METHODOLOGIES(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF SYSTEMS AND LOGISTICS 1/2

UNCLASSIFIED S D BRANCH DEC 87 AFIT/GIR/LSR/87D-1 F/G 5/2 NL




1·0
1·1
1·25
1·4
1·6
1·8
2·0
2·2
2·5
2·8

(2)

A COMPARISON OF INFORMATION SYSTEM DEVELOPMENT METHODOLOGIES

THESIS

Steven D. Branch
Captain, USAF

AFIT/GIR/LSR/87D-1

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

2

AFIT/GIR/LSR/87D-1

A COMPARISON OF INFORMATION SYSTEM DEVELOPMENT METHODOLOGIES

THESIS

Steven D. Branch
Captain, USAF

AFIT/GIR/LSR/87D-1

The contents of the document are technically accurate, and no sensitive items, detrimental ideas, or deleterious information is contained therein. Furthermore, the views expressed in the document are those of the author and do not necessarily reflect the views of the School of Systems and Logistics, the Air University, the United States Air Force, or the Department of Defense.

AFIT/GIR/LSR/87D-1

A COMPARISON OF INFORMATION SYSTEM DEVELOPMENT METHODOLOGIES

THESIS

Presented to the Faculty of the School of Systems and Logistics

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Information Resource Management

Steven D. Branch, B.B.A.

Captain, USAF

December 1987

Acknowledgements

The author wishes to express sincere appreciation to several individuals who provided guidance and inspiration throughout this effort.

First and foremost, I thank my wife, Lynna Ann, for understanding me so well throughout this last year. Without her love and encouragement this project would never have been finished.

Secondly, I thank Capt Ken Jennings who provided such thoughtful and expert guidance. Ken proved himself to be an expert in both organizational development and human relations. How else could he have put up with me for this entire year?

Finally, thank you to Dr. Charles Fenno and Lt Col John Ballard. Both of these men are professionals in every sense and served as an inspiration to us all.

Table of Contents

List of Figures

List of Tables

AFIT/GIR/LSR/87D-1

Abstract

This research critically evaluated a select sampling of current information systems development methodologies. The research had two primary objectives. The first was to enhance the manager's understanding of information systems and design methodologies. The second was to provide a tool to assist managers in deliberately choosing which methodology best fits their own specific needs.

Two sets of attributes were selected as the basis for an evaluative framework to compare methodologies. One set was chosen based on an information systems development life-cycle model. The intent was to specifically compare the utility of the methodologies for development throughout the complete range of information systems development.

Another set of attributes was derived from factors that contribute to the institutionalization of the change represented by the information system in the organization. Each of the selected methodologies was then compared in relation to its degree of coverage of the attributes in the framework. A matrix format was used to present the relative coverage of the different methodologies within the bounds of the framework.

The research led to several conclusions. First, structured methodologies have tended toward a narrow focus in support of logical and physical design rather than expanding

into a broader framework including proactive management of the change process. Second, socio-technical methodologies inherently paid more attention to factors which provide both a complete requirements analysis and support for the institutionalization of the change process. However, results from their use provide few specifics which can be easily translated into program code. Finally, since computer aided software engineering (CASE) methodologies appear most promising in their ability to allow development efforts to focus on the actual problem at hand vice the complex aspects of the solution to the problem, a merger of CASE with Socio-technical approaches is recommended.

A COMPARISON OF INFORMATION SYSTEM DEVELOPMENT METHODOLOGIES

I. Introduction

Background

Due to dramatic technological improvements and reduction in costs, computers are no longer confined to their more traditional roles of historical accounting and data processing activities (2:12, 37:1). In fact, the idea that management information systems (MIS) are useful is clearly supported by current literature. If a carefully designed and implemented management information system has a good fit with organizational needs, it can, and in many cases does, improve the manager's decision making process (14:28-54, 21:64-67).

More and more organizations, having realized the value of information as a resource, are designing and implementing management information systems unique to their own specific needs. With that in mind, it is logical to assume that increasing numbers of managers will have to make decisions concerning the design and implementation of these systems.

There are many different systems development tools and techniques in use today. However, there is a lack of consistent information concerning the relative attributes of these methods even among professional systems developers (12:51). Colter describes the confusion related to this

information deficiency in his comparative examination of techniques:

> They [methodologies] are not clearly understood by many practicing professionals. They tend to be incomplete, requiring evaluation and integration to result in coherent analysis processes. Unfortunately, the literature is void of any work which could aid this integration process. (12:51)

Maddison's rationale for the scarcity of information regarding the relative features of different methodologies is that "most people's experience is limited to a single methodology" (29:1). He implies that most systems developers, after having become familiar with one methodology, will tend to continue to use that methodology. Moreover, since becoming proficient in the use of any particular methodology is a fairly complex task, the practicing systems developer probably does not tend toward experimentation with other methodologies (29:1).

Specific Problem and Justification

The primary goal of this research is first to critically evaluate current methodologies and synthesize the information gained from the evaluation. Then, as a result deriving from this work, a guide will be developed that should be useful to managers in both understanding information systems and in selecting methodologies suitable to their needs. Achieving these goals will be done in several steps.

First, since this research is primarily aimed at managers, it must provide some conceptual foundations to enhance their understanding of information systems and design methodologies. A brief background in current thought concerning the application of information systems is provided to help the manager take a more proactive role in the process of information systems change. It naturally follows that the manager with good understanding of how technology can affect the structure and the processes of the organization is better equipped to manage the changes that result from the addition of information systems to the organization.

Second, this research should result in a useful guide to help managers determine which methodologies best fit their own specific needs and circumstances. Choosing the right methodology is a complex task which can be critical to the success of the information system (34:49, 16:179, 12:51, 14:444, 18:1631). A guide for comparing methodologies should be a very useful management tool since, at present, the resources available for making such a choice are scarce (12:51).

Scope and Limitations

There are many methodologies currently in use throughout the world. Since an attempt to discuss and evaluate each one would be impractical, this research has focused only on a representative sampling of the

methodologies in existence. The intent of this research was to present examples of methodologies which current literature portrays as credible, practical, and widely used.

The goal of this project was to make design methodologies more understandable and effective for the manager. Consequently, a framework for evaluation was sought that would be both simple and comprehensive. Its understandability, however, should neither require an extensive MIS background nor an in-depth familiarity with technical terminology.

The model chosen to compare methodologies is shown in figure one. In this model the effectiveness of a methodology is equal to the sum of its development life-cycle coverage and its support for the institutionalization of the change process. The operationalization of this model will be developed later.

Life-Cycle Coverage	+	Support For Institutionalization	=	Effectiveness

Figure 1. Model of Development Methodology Effectiveness

II. Methodology

General Method

The method used in this research consisted of a comparative analysis based on review of the literature and direct qualitative comparisons on selected attributes.

The following types of sources were most helpful in obtaining relevant information:

1. Books: Specialized textbooks were a particularly useful source of information regarding conceptual foundations and a general description of methodologies.

2. Journals: Journal articles were a good source of information for a more specific description of methodologies.

3. Conference Proceedings: The documentation from the IFIP WG 8.1 Working Conferences known as CRIS1 and CRIS2 (Comparative Review of Information System Design Methodologies) was most helpful in this work (6, 7, 19, 40).

4. Vendor literature.

5. Personal interviews with experts in the field.

To assist the reader in determining the dimensions of this study, a tabular presentation of the types of references used in this research is depicted in table I.

Research Objectives and Activities

The following objectives and activities provide the strategic guideline for completing this research. The objectives state the task to be performed and the

TYPE OF REFERENCE	MAIN ORIENTATION OF REFERENCE	# REFERENCES
JOURNALS	MIS	19
	Management	3
BOOKS	General MIS Books	8
	Socio-Technical MIS	3
	MIS Methodologies	6
BROCHURES	Product	1
TOTAL REFERENCES		40

Table I. Characterization of Selected Bibliography

corresponding activity describes the method used to accomplish each task.

Objective One. Broaden the manager's understanding of the application of information systems to the organization.

Activity One. Through an analysis of the literature, a conceptual foundation for understanding the general applications of information systems to organizations was presented. The topics chosen for discussion were those judged most relevant from the manager's perspective rather than the systems development perspective.

Objective Two. Develop a workable typology of information system development methodologies.

Activity Two. This activity entailed determining how the various methods could be categorized into groupings of similar types. A typology was formed using the primary

orientation of methodologies as the main differentiator. This method allowed a relatively well-defined classification of the methodologies. It also provided a means to group methodologies into meaningful categories from the manager's viewpoint.

Objective Three. Develop a framework which provides a tool to measure the effectiveness of methodologies in relation to each other.

Activity Three. Two sets of attributes were chosen to make up the evaluative framework. One set was chosen based on an information systems development life-cycle model. There was considerable disagreement among several well known authors as to the stages necessary in the information systems development life-cycle. Therefore, a generic life-cycle model was generated in an attempt to combine the best ideas from each.

The second set of attributes was derived from factors that contribute to the institutionalization of change in the organization. This additional set of factors was intended to keep the perspective of the research meaningful to the manager and lend further criteria to measure the effectiveness of methodologies. Rationale for the selection of these two particular sets of attributes will be presented later.

Objective Four. Compare each methodology in relation to its degree of coverage of the attributes in the framework.

Activity Four. Using the literature mentioned earlier, methodologies were described and compared within the framework developed in activity three.

Objective Five. Present the results of this study in a manner useful to the manager.

Activity Five. A matrix format was used to display the relative coverage of the different methodologies within the framework. This format should enhance the manager's capability to compare and analyze the effectiveness of one methodology relative to another within the bounds of the framework.

Now that the objectives and activities of this research have been stated, the following chapter will begin the process of building a conceptual foundation for understanding the application of information systems to organizations.

III. Analysis of the Literature

Introduction

This chapter will introduce some important fundamental concepts and build a typology of information systems development methodologies. The first section lays a conceptual foundation for understanding the basic application of management information systems to organizations. The second section, which classifies the methodologies, begins with a discussion several differing opinions by well known authors. It then presents the typology chosen for this research. The typology will be used later when the relative attributes of different methodologies are discussed and analyzed.

Conceptual Foundations

Definition of Management Information System (MIS). The definition of the term "Management Information System" continues to change with the evolution of the capabilities of computer systems. In the early seventies, Mason and Mitroff defined an information system as:

> At least one person of a certain psychological type who faces a problem within an organization context for which he needs evidence to arrive at a solution (i.e., to select some course of action) and that the evidence is made available to him through some mode of presentation. (33:2)

Later in that same decade, Jenkins defined MIS as "at least one person utilizing an information system to undertake a task and the resulting performance" (33:2). It

is interesting to note the absence of the word "computer" from the previous definitions. Nolan and Wetherbe suggest that they are effective in a "micro" sense because they contain the minimum requirements for an information system. However, they feel that a broader definition is required to address a wider range of issues concerning MIS such as its affect on "organization structure or organizational processes" (33:3). The following definition given by Davis and Olson will serve as a definition of MIS in this research effort:

> A management information system . . . is an integrated, user-machine system for providing information to support operation, management, analysis and decision-making function in an organization. The system utilizes computer hardware and software; manual procedures; models for analysis, planning, control and decision making; and a database. (14:6)

Davis and Olson explain that one can, of course, have an MIS without computers; but, the "power of the computer is what makes MIS feasible" (14:7). Further, the concept of a user-machine system is an important lead-in to the idea that some tasks are best performed by humans and others best performed by machine (14:7).

Definition of Methodology. The meaning of the term "methodology" varies widely from one author to another in the literature. The core of a definition for this paper is one given by Maddison:

> An information system methodology is a recommended collection of philosophies, phases, procedures, rules, techniques, tools, documentation,

> management and training for developers of information systems. (29:4)

One important clarification of this definition is necessary. The term "developers of information systems" does not solely refer to information systems professionals. This research assumes that any person in the organization can take part in the development of an information system.

Levels of Management and Control. A classic framework for viewing MIS from its capability to support management decision making is suggested by Gorry and Scott-Morton (21:55-70). They assert that since information systems exist mainly to support decision making, it is appropriate (especially from an information systems point of view) to characterize the organization from the standpoint of the types of decisions involved at different levels. Their framework (based on a previous model suggested by R. N. Anthony) classifies organizational activity into three different levels; strategic planning, management control, and operations control (see figures 2 and 3). The implications of this model to information systems analysis and design become evident as the applicability of different methodologies are compared to different levels of management.

Strategic Planning. The top level, Strategic planning, is the process of deciding on the goals and objectives of the organization (21:57). It is characterized typically as involving a small number of people who must

make nonroutine and creative decisions (21:57). Since the main concern at this level is predicting the future of the organization and anticipating changes to its environment, the decision making process contains the most uncertainty and the least structure (21:57).

Management Control. The middle level, management control, is concerned with such problems as the acquisition of resources, the establishment and monitoring of budgets, and the development of new products (21:57). It is most often concerned with people (21:57). Decision making at this level is best characterized as semi-structured under conditions of medium uncertainty (21:57).

Operational Control. In the bottom level, operational control, the focus is on effective and efficient use of existing resources (21:57). Management at this level most often deals with the accomplishment of tasks within the constraints of existing resources (21:57). The characterization of decision making at this level is high structure with low levels of uncertainty (21:57).

The main reason for presenting this model is to point out to the manager that certain methodologies are suited to specific levels of management control. For example, a methodology designed for strategic level analysis may be well suited for use at the strategic management level of control (high uncertainty, low structure) while at the same

time not useful at the level of operational control (low uncertainty, high structure).

Furthermore, Davis and Olson even suggest that the best underlying rationale for determining an appropriate development method is the degree of uncertainty surrounding both the decision making process and the development process (14:488). This implies that knowing which methodologies are best suited to which levels of uncertainty can enhance the manager's ability to match a given task to an appropriate methodology.

Strategic Planning	Management Control	Operations Control

Level of Uncertainty

High <-------------------------> Low

Figure 2. Relationship between Organizational Levels and Uncertainty

Strategic Planning	Management Control	Operations Control

Amount of Structure

Low <------------------------------> High

Figure 3. Relationship between Organizational Levels and the Amount of Structure in the Decision Making Process

<u>Control</u>. Information systems are often used to report variances from a standard. This is the essence of control. As Davis and Olson explain: "The purpose of organization and control is to reduce uncertainty regarding the task to be performed, how it is to be performed, and when it will be performed" (14:321). This concept is related to the previous discussion of management level of control. If control is a necessary ingredient in a proposed information system (as opposed to a decision support system for planning only), then having a knowledge of the level of management and decision making involved is important. This knowledge helps to determine the types of control needed and, hence, gives clues as to the type of information system required.

Classifying the Methodologies

Introduction. A close scrutiny of the literature reveals much disagreement about the classification of systems development methodologies. Some authors feel that as few as two categories are sufficient (12:51) while others suggest that as many as eight are required (14:483). This section presents several different views and then gives the typology decided upon for use in this research.

A Two Category Opinion. Colter feels that two categories, traditional and structured, can classify analysis techniques. He states that traditional analysis "concentrates on input, output, and processing detail" (12:52). On the other hand, he says that structured analysis "concentrates on the various structural aspects of systems" (12:52).

A Three Category Opinion. According to De and Sen, three categories are sufficient to classify all methodologies: data analysis, decision analysis, and activity analysis (16:179). They classify data analysis as the "traditional bottom up approach" which examines the processes currently being used by the organization and develops the information system to mimic those processes (16:180).

In their view, decision analysis is mainly concerned with the decisions being made at different management levels of the organization. As stated by De and Sen:

> Typically, the decision analysis approach is supported by those who hold the belief that decisions define all information requirements, and that an effective design is only possible if a model of the decision process is developed first. (16:180)

Finally, they feel that activity analysis includes those methodologies that tend to define information requirements in accordance with the Gorry and Scott-Morton model previously described. De and Sen explain:

> The information needed by the strategic planners is aggregate; the scope and variety of this information varies extensively. By contrast, the information needs of operations people are well defined, narrow in scope, and require detailed statements. The information requirements for management control fall between those of strategic planning and operations control. (16:180)

De and Sen assert that their typology is adequate to categorize methods for requirements determination. However, it is not suitable for this research because it fails to provide sufficient classification to provide for all of the methodologies found by this research. A more appropriate classification scheme might be one similar to that of Davis and Olson which concentrates on primary orientation of the methodology as the main factor.

A Primary Orientation Approach. Based on the primary orientation of each methodology, Davis and Olson (14:482-488) suggest eight approaches to information requirements analysis:

1. Normative analysis
2. Strategy set transformation
3. Critical factors analysis
4. Process analysis
5. Ends-means analysis
6. Decision analysis
7. Socio-Technical analysis
8. Input-process-output analysis

Each of these methods has merit as described in the literature even though some of them are not supported by specific methodologies. For example, they describe normative analysis as a method which is "based on the fundamental similarity of classes of object systems" (14:483). In other words, if there is a basic set of requirements associated with a general type of application (i.e. accounting department, inventory control) then this strategy would concentrate on tailoring the fundamental "set" of requirements to a specific organization or application. Even though no specific methodologies were found in the literature to support "normative analysis," this mode of operation can be a very effective and powerful method of developing an information system.

Of the seven other classifications given by Davis and Olson, two are focused on organizational goals and objectives. These include strategic analysis and critical factors analysis. Four of the five remaining: process analysis, ends-means analysis, decision analysis, and input-process-output analysis, can be grouped into a category called "structured analysis techniques." Finally,

socio-technical analysis falls into a unique category of its own..

Additional Categories. Three additional categories are worth mention due to their emphasis in current literature: prototyping, computer aided software engineering, and end-user development.

Prototyping. Prototyping is a relatively new approach to systems development based on the idea that it is best to quickly give the user a model with which to work (14:568). Davis and Olson describe the concept:

> The prototyping methodology is based on the simple proposition that people can express what they like or do not like about an existing application system more easily than they can express what they think they would like in an imagined, future system. (14:568)

With this approach, once the initial model is presented, the analyst can proceed to fine-tune it using an iterative process until both the analyst and the user agree that the design fits the user's needs.

Computer Aided Software Engineering (CASE). Computer aided software engineering is not a new concept; however, recent advances in hardware and software are bringing the automation of the systems design process closer to reality. Through the use of "fourth generation" languages, some advocates believe that it is now possible to produce "rigorous computable specifications and then automate the generation of program code" (30:37).

User-Developed Applications (UDA). Finally, the category of user-developed applications is one which might not seem (at first glance) to have a rightful place in this discussion. However, it is the contention of this researcher that UDA should not be ignored for two reasons. First, UDA cannot be ignored because effective user-developed applications do exist and users are generally becoming more and more sophisticated (35:90). Further, managers are increasingly faced with decisions regarding the applicability and management of user-developed applications (14:613).

The Typology for this Research. The following list of development approaches will form the typology for use in this research:

1. Strategic analysis
2. Critical factors analysis
3. Structured analysis
4. Socio-Technical-Systems analysis
5. Prototyping
6. Computer aided software engineering
7. End-user development
8. Normative analysis

This typology is conceptually the same as Davis and Olson because the primary orientation of the methodology is the main consideration. However, in addition to their typology, prototyping, computer aided software engineering, and end-user development have been included as new categories to allow for discussion and comparison of a full range of concepts and methodologies important to this

research. As discussed earlier, all of the structured analysis techniques have been grouped into one category.

Except for normative analysis, which requires no further discussion, the next section will give a summary of each approach. An example of one specific supporting methodology will be included in the discussion (where applicable). In cases where more than one specific supporting methodology per classification is included (e.g. structured analysis and socio-technical-systems analysis), descriptions of the additional specific methodologies can be found in the appendix. In the case of prototyping and end-user development, no specific supporting methodologies were found. Therefore, the reader is given a description of the general category in detail.

Overview of Methodologies

Strategic Analysis. Strategy Set Transformation (SST) is the only method found by this literature search which was aimed specifically at information systems planning at the strategic level. The SST approach, developed by W. R. King, views the organization as an 'information set' containing the mission, objectives, strategies, and other strategic variables (5:16). Davis and Olson describe it as a method for 'alignment of the information system plan with organizational objectives' (14:483).

Description. The use of Strategy Set Transformation is summarized in the following steps:

1. Identify the organizational strategy set.

    a. Delineate the organization's claimant structure. A claimant is someone with a valued interest in the organization such as owners, managers, stockholders or suppliers.

    b. Identify goals for each claimant group.

    c. Identify organizational purposes and strategy relative to each claimant group.

2. Present tentative statement of organizational goals and strategies to top management for review and comment.

3. Transform the organizational strategy set into an MIS strategy set.

    a. Identify the MIS strategic elements for each element in the organizational strategy set.

    b. Identify information system constraints and objectives.

    c. Identify information system design strategies based on organizational attributes, information system constraints, and information system objectives. (14:459)

<u>Discussion</u>. Bowman et al. note that one disadvantage of the method is that it focuses exclusively on strategic MIS planning. They also point out that extensive manager/user input and review is required to achieve "accurate and concise articulation of organizational objectives and strategies" (5:17).

<u>Critical Factors Analysis</u>. The primary orientation of this approach is to determine the set of factors that managers deem critical to the success of the organization. A good example of this technique is the "Critical Success Factor" (CSF) method developed by J. F. Rockart (36:81-93).

Once critical factors are identified with this method, they can then be stated as information systems objectives.

Description. The CSF method is usually conducted as a series of no more than three interviews with organization personnel (36:85). First, the executive's goals are determined and discussed for clarification. Second, there is usually a session for review and "sharpening up" of the factors by the analyst. Finally, a third session may be used to obtain final agreement (36:85).

Rockart defines CSFs as follows:

> Critical success factors thus are, for any business, the limited number of areas in which results, if they are satisfactory, will ensure successful competitive performance for the organization. They are the few key areas where 'things must go right' for the business to flourish. (36:85)

Rockart notes that there are four prime sources of critical success factors:

1. Structure of the particular industry. For example, to stay competitive, supermarket chains will have to pay attention to different CSFs than an automotive industry.

2. Competitive strategy, industry position, and geographic location. Each different company in an industry will have its own unique situation determined by these three factors.

3. Environmental factors. This concerns factors such as the state of the economy or the cost of fuel.

4. Temporal factors. These are factors that may appear as CSFs due to unusual circumstances and are usually temporary in nature. Rockart uses the example of having a group of executives die in a plane crash. An accident like this might create a temporary CSF to rebuild an executive group (36:85).

Discussion. Rockart points out several benefits of the CSF approach to the general manager (36:88). He says that it helps managers determine where to focus attention, aids in developing good measures for the factors, helps clearly define the size of information requirements, and is a significant aid to the planning process. However, he stresses that the CSF method is not a strategic planning method. Instead, it focuses on information needed for management control (36:88).

Rockart suggests that this method is easy to explain to executives and that feedback concerning the process and its outcome has been good (36:85). On the other hand, Kotteman notes that CSFs can be quite subjective. He warns that the process can lead to an inconsistent set of requirements if not used carefully (26:54). According to Shank et al., the CSF method has been successfully used in information resource planning and (contrary to Rockart's discussion) strategic planning (39:127). It is applicable to both the organization and application level (14:485).

Shank et al. describe several interesting outcomes from one experiment with critical success factors (39:127). First, once all staff members understood and accepted the organization's CSFs, acceptance of the MIS plan (developed from those objectives) was good. Second, the "intuitively appealing" nature of the methodology caused early acceptance of senior level management. Third, the methodology

developed a "core of information technology proponents throughout the organization and enhanced the understanding of MIS by management." Finally, they mention one interesting positive spin-off. The process of identifying critical success factors gave all staff members a better understanding of the broad goals and activities of the organization and helped individuals as well as departments align their goals and objectives with those of the corporation (39:127).

Structured Analysis Methods. This category of methodologies is primarily concerned with a top-down, structured approach to systems analysis and design. Colter traces the roots of these methodologies to problems emerging in the 1960s (11:73). As computer systems evolved into more complex combinations of hardware and software, "there was a general agreement that our ability to manage the software development process could not meet the need for increasingly complex systems" (11:73).

Colter describes the general goal of the structured method as "the development of systems that meet user requirements through an orderly and manageable process" (11:75).

Structured analysis techniques include the use of a variety of tools to aid the analyst in developing a systems model. These tools include, but are not limited to: data flow diagrams, HIPO charts, functional decomposition,

Jackson charts, Warnier-Orr diagrams, and pseudocode (11:85-89). According to Colter, these tools are "a set of graphic techniques that both assists the design process and represents the design at various levels" (11:87). A comprehensive overview of most structured analysis tools can be found in Tools and Techniques for Structured Systems Analysis and Design by William S. Davis (15).

In general, the structured methodologies vary widely in terms of which area of the systems development cycle receives focus. Some focus on requirements analysis while others concentrate on design guidelines (11:92).

Business Systems Planning. Perhaps one of the most comprehensive of the structured analysis methodologies is IBM's Business Systems Planning (BSP). BSP was originally developed by IBM for their own internal use. However, IBM customers expressed enough interest in learning how to manage their computer resources that it was made available to the public (5:17).

Description. According to IBM, the basic doctrine of the methodology is described as follows:

> A fundamental tenet underlying BSP is that an information systems plan for a business must be integrated with the business plan and should be developed from the point of view of top management and with their active participation. (24:237)

BSP is primarily a two phase process consisting of an identification phase and a definition phase (24:237). The main goal of the identification phase is to understand the

business. Then, the definition phase takes the information systems network derived from the first phase and turns it into a detailed plan for designing and implementing the information system (24:237).

IBM states that BSP is based on the following three principles:

1. Establishment of a business-wide perspective.

   This principle is specifically oriented toward identifying and defining the planning and control of the business problems of general management (24:242). (The framework used is very similar to the Gorry and Scott-Morton Model)

2. Top-down analysis, bottom-up implementation.

   a. The concept of top-down analysis is supposed to ensure that business needs and priorities in defining the system maintain a top management perspective.

   b. Bottom-up implementation relates directly to business processes necessary to achieve the objectives of the business.

3. Systems and data independence.

   The main concern of this objective is to define systems to be as independent of specific organizations as possible. IBM says the key to providing organizational independence lies in the identification and definition of the business processes. These two activities make up the primary phases of the BSP approach. (24:245)

<u>Discussion</u>. Bowman et al. note two potential drawbacks of the methodology. First, even though they agree that the approach can be effective in identifying current requirements, they warn that careful consideration must be given to overall strategic planning to ensure that the

resulting plan has a "proper long-range perspective" (5:18). Second, they caution that BSP's comprehensive nature (involvement of many managers and requiring the synthesis of voluminous data) can make it difficult to come up with a "viable information system" (5:18). Davis and Olson concur that BSP is a comprehensive methodology and state that it is "well supported by materials and instruction" (14:485).

Socio-Technical-Systems Approach (STS). As discussed by Bostrom and Heinen (3:17), this approach is based on the belief that organizational behavioral problems are the prime cause of many MIS failures. The STS approach assumes that the organization is made up of two "jointly independent, but correlative interacting systems - the social and the technical" (3:17). Bostrom and Heinen recognized the argument that technology was a necessary evil; however, they disagreed with the proposition that it was solely up to the people of the organization to adapt:

> Our basic premise is that computer-related technology is essentially neutral; whether its application succeeds or fails depends entirely on the decisions that are made on how it shall be used. (3:18)

The general STS design approach presented by Bostrom and Heinen involved three phases which were:

Phase I - Strategic Design Process.

Involves explicit formulation of the goals and objectives of the project.

Phase II - Socio-Technical System Design Process.

Emphasizes both the procedural aspects of design and the change process of the social system.

Phase III - Ongoing Management Process.

Involves a continual monitoring of the system. Views implementation as an iterative process of fine tuning. (4:17)

Mumford's ETHICS, a socio-technical methodology developed over a period of 15 years (23:111), will be presented in this section. Another socio-technical methodology, developed by Pava, is similar to ETHICS but applies the socio-technical approach more to the domain of the office (23:125). A description of Pava's methodology is in the appendix.

Mumford's ETHICS. As reported by Hirschheim, the ETHICS methodology consists of the following six stages:

1. Essential systems analysis
2. Socio-technical systems design
3. Set alternative solutions
4. Set compatible solutions
5. Rank socio-technical solutions
6. Prepare detailed work design from chosen solution (23:112)

Discussion. The key to this socio-technical systems approach is viewed by Hirschheim as being "its participative nature" (23:111). He says that while users play an important role in the development of any information system, user involvement is essential to the nature of

ETHICS. However, he says that the participative nature of the methodology should not be overemphasized as it is in some of the literature. He feels that ETHICS attempts to operationalize the socio-technical philosophy and that participation is only one factor among many that are involved in that process (23:111).

Bostrom and Heinen believe that the socio-technical method is a plausible solution for many of the failures of systems implementations caused by the traditionally narrow way "systems designers view organizations, their members, and the function of an MIS within them" (3:17). Further, they feel that the use of the STS approach can greatly reduce the number of MIS failures (3:17).

On the other hand, Paddock points out that the division of development into technical and social systems, "with the associated need for a behavioral scientist/OD [organizational development] professional, may increase the level of conflict present and/or shift its focus" (34:54). Moreover, he points out that the user and designer roles may be changed by the STS process:

> . . . it is conceivable that in attaining acceptance, the user's customary role as co-negotiator with the designer under a traditional model could evolve into one of mediator between MIS and OD professionals in accommodating technical and social goals and options. This role shift may be undesirable from the user's standpoint, causing them undue pressure by calling for more knowledge than they have, putting them at a disadvantage with both the MIS and OD professionals. (34:55)

Paddock goes on to say that if the role shift is significant then enhanced training for the users may be required to help them function more effectively in their new roles (34:55).

Prototyping. As previously mentioned, the primary goal of the prototyping method is to get a workable model into the hands of the user quickly so that the user and analyst can work together in an iterative fashion until the design fits the user's needs (14:568).

Description. Naumann and Jenkins (32:31-33) view the process of prototyping as a four-step procedure:

> Step 1 - Identify the user's basic information requirements.
>
> The analyst may use either a data abstraction approach (such as a database driven pilot system) or a process modeling approach. In either case, Naumann and Jenkins argue that completeness is not important at this stage.
>
> Step 2 - Develop a working prototype.
>
> Developing the working simulation quickly serves both the user and the analyst. The user has a tangible model to evaluate and the analyst receives responses based on the user's evaluation.
>
> Step 3 - Implement and use the prototype system.
>
> The prototype model exploits the user's ability to find problems and irritants with the system through the iterative process of development.
>
> Step 4 - Revise and enhance the prototype system.
>
> This step requires identifying and correcting the problems the user experienced in step 3. Rapid turnaround remains important. Steps 3 and 4 continue to be repeated until the user accepts the system.

Cerveny and others agree with Alavi's definition of prototype:

> A prototype is a real, working, and usable system built economically and quickly with the intention of being modified (1:19)

They also assert that there are two factors that impede communication between the analyst and the user in the traditional approach:

1. The abstract tools used in the system development process.

2. The concurrent learning process of the user and analyst during system design. (10:54)

According to Cerveny and others, the communication process is hampered by the lack of an appropriate medium to exchange ideas because flow charts, file layouts, and relational data diagrams are difficult for the average manager to comprehend (10:54).

They suggest a framework consisting of three levels of prototyping which blends nicely with previous discussion of management levels of decision making:

> <u>Level 1</u> - Input/Output Design.
>
> Generation of printed reports or on-line screens. Not concerned with interactions of data or relationships among files and transactions. Its main objective is facilitating communication between users and systems developers while producing a superior form of design documentation. (10:59)
>
> <u>Level 2</u> - Heuristic Design.
>
> Includes the design of systems functions. Use of a relational database. Minimizes system development time and effort. Level two

prototyping does not advocate the development of a complete system. (10:60)

Level 3 - Adaptive Design.

Level three involves the complete development of a prototype system which is maintained in a prototype state to allow for an evolutionary design throughout the project's useful life. (10:60)

Cerveny et al. recommend level one prototyping for transaction processing. The main function at this level is to capture and retain organizational transactions and to provide simple reports and query capabilities (10:59). This would be a highly structured situation with a low amount of uncertainty (10:59).

Level two prototyping is described as providing management with information on how efficiently organizational resources are being utilized (10:59). Systems are part of the organization's control mechanism. This level has more design uncertainty than level one. According to Cerveny et al., as design uncertainty increases, the advantages of a more complete prototype increase (10:59). In addition, they suggest that as the level of uncertainty rises, so does the effectiveness of the prototyping approach (10:61).

This argument leads to the conclusion that prototyping may be most effective at the top level of the organization (strategic planning). This level contains the most uncertainty and involves the least programmable decisions.

This is also the level most applicable to the development of decision support systems (10:60).

Discussion. According to Naumann and Jenkins, prototyping can be applied at any organizational level. However, they agree with Cerveny et al. that it will probably be most useful in those areas where there is less stability and more uncertainty in the decision making process (32:37).

Cerveny et al. address issues related to the implementation and function of prototyping in the traditional systems development life-cycle. They maintain that "the purpose of the prototype is to facilitate interaction and learning by the user and the analyst" (10:53). Further, they argue that prototyping is needed because the traditional life-cycle development approach fails to adequately consider the issue of poor communication between the user and the analyst:

> Perhaps the most important reason for prototyping's effectiveness is the possibility that it can foster a climate of positive attitudes and constructive conflict between the user and the analyst (10:55)

Computer Aided Software Engineering (CASE). According to Konsynski, the ultimate goal of software engineering is the "formalization and automation of the system development process" (25:11). Further, he notes that most of the current software engineering activity centers on the following aspects of systems development:

- Providing tool environments to aid in the development of more reliable systems

- Building generalized systems and systems for "virtual" environments

- Making more use of existing code (logic and designs) - reusability

- Instituting production management techniques and configuration management practice

- Reducing the maintenance costs by building maintainable systems

- Building tools for end-user development

- Exploiting hardware cost reductions through task distribution in networked processors

- Using artificial intelligence techniques in software design and development (25:24)

The common thread among CASE methodologies is the fourth-generation or very high level development language. Some of the desirable features associated with fourth-generation languages are described by Davis and Olson as follows:

- Interactive dialog to guide application development

- Simple to learn with helpful error messages

- Relational database management

- High-level query language for direct access to the database

- Graphics capabilities

- Interactive editor for interactive update and retrieval

- High-level instructions that reduce the number of program statements required (14:424-425)

James Martin says that, due to the proliferation of computers, "it is essential to be able to develop applications with far less manpower" and this means the "automation of automation" (30:19). Moreover, Martin contends that productivity gains of 1000% or more are not uncommon with the use of data-base user languages, report generators, graphics packages, and application generators (30:23).

Information Engineering (IE). Martin's "Information Engineering" approach will be used in this research as an example of a CASE methodology.

Description. Martin and Hershey describe the methodology as the following four stages:

> Stage 1 - Information Strategy Planning.
>
> Concerned with top management goals and critical success factors, a high-level overview of the enterprise, its functions, data, and information needs.
>
> Stage 2 - Business Area Analysis.
>
> Concerned with what processes are needed to run a selected business area, how these processes interrelate, and what data are needed.
>
> Stage 3 - System Design.
>
> Concerned with how selected processes in the business are implemented in procedures, and how these procedures work. Direct end user involvement is needed in the design of procedures.
>
> Stage 4 - Construction.
>
> Implementation of the procedures using, where practical, fourth-generation languages, code generation, and end user tools.
> (31:8)

Discussion. Martin and Hershey profess that their four-stage engineering process concentrates much more time on planning and design than on execution and that the use of advanced automation techniques makes this possible. Further, they describe the key objective of information engineering as: "to impose rules on planning and design that are formal enough to direct the computer to write code, thus freeing the MIS professional from the burden of coding" (31:14).

Konsynski heartily agrees with Martin and Hershey on the use of fourth-generation languages and methods:

> Many vendors of Fourth Generation tools claim that these techniques are designed to generate solutions fast - at least ten times faster than a third generation language such as cobol. The reality is that under certain applications and certain environments, many of these tools do perform faster than in second generation environments. Speed, however, is not the major motivation for acceptance of these tool environments. They not only support access to information but also help to analyze, model, and present information in a form understandable by users. (25:25)

What is more, Konsynski concurs that fourth-generation languages change the focus from "creating a computer solution" to solving the actual problem at hand. He says that this shift in focus means more time can actually be spent on solving business problems vice writing computer code (25:25).

In summary, Martin and Hershey claim that the important characteristics of information engineering are as follows:

- Driven by the user
- Based on easy-to-understand diagrams
- Requires full automation
- Links design automation to code generators and fourth-generation languages where practical
- Uses prototypes
- Assists information center activities
- Achieves fully integrated organizational data processing (31:38)

<u>End-User Development</u>. During the last two decades, improvements in the "cost/performance ratio" of the hardware supporting the computing industry have averaged 30 to 40 percent per year and are expected to continue at this rate well into the 1990s (2:12). Benjamin predicts that because the terminal may be as common as the telephone by the 1990s (in the workplace), the end-user will dominate as much as 75 percent of all available computing resources (2:17).

The growth in end-user computing is significant. It has risen from an estimated 2.6 million in 1982 to 5.6 million in 1984 and is projected to continue to grow to an estimated 13 million by 1990 (22:179).

The increase in end-user computing is not only causing an increase in numbers of terminals, it is also changing the structure of the information systems environment within the organization. As more and more end-users are interactively involved in computing, they are becoming less dependent on the data processing (DP) department to provide computing

services and, instead, more demanding of the management expertise of the DP professional (2:24). Rockart and others say the future will hold "increasingly computer knowledgeable and demanding users" for two reasons. The first is because the college graduates of the future will believe the computer is a necessary tool. The second is because there will be an increased general understanding of computers by the work force through their association with home computers (37:2). One evidence of change and decreased dependency on the DP department is a relatively new phenomenon called "user-developed applications."

<u>User-Developed Applications</u> (UDA). In the past, accepted protocol allowed the user to do little more than make suggestions about applications development (35:90). However, since the typical DP department may be "months or even years behind schedule" in keeping up with demands for applications software, many users are developing their own applications (35:90). Rockart and others suggest that future managers will have to "provide the newly sophisticated end-users with the automated tools which they are ready for and willing to use" (37:2).

Rivard and Huff note that some of the new and bright end-users are developing quite sophisticated applications:

> While most user developed applications are small, some did require more than three months of effort. Many DP managers may find it surprising that users are developing systems requiring such an effort. (35:97)

As Carey and Young describe the necessary ingredients required to successfully integrate the personal computer into the workplace, they stress that there must be an increased emphasis on end-user training (8:35). Many top information executives agree that the "facilitation and management" of end-user computing is one of the biggest challenges facing the information systems staff of the future (17:137).

Description. Konsynski discusses end-user development from two aspects (25:24-29). First, he notes that fourth-generation languages and other "state of the art" tools are often able to change the focus to solving the business problem vice constructing a computer solution to the problem (similar to Martin's previous argument in favor of fourth-generation languages).

Secondly, he promotes the concept of the Information Center. The Information Center is a central facility which contains hardware, software, training, and consulting to assist end-users (14:427). Konsynski feels strongly that the Information Center is a necessary ingredient in the effective implementation of end-user development. He compares its potential impact to that of IBM's highly successful computer network architecture, Systems Network Architecture (SNA):

> In the final analysis, the Information Center concept will do to end-user computing what SNA did for the evolution of data communications support. This is to say that it provides a framework and a

migration strategy for developing the end-user capability in a controlled, phased fashion. (25:29)

Discussion. Konsynski notes that, at present, there is much skepticism concerning the value of the Information Center approach to managing end-user computing. However, in his opinion, the skepticism will subside as the proliferation of end-user computing generates a need for more guidelines in this area (25:29).

Davis and Olson note several advantages of user-developed applications:

1. Relieves shortage of system development personnel.

2. Eliminates the problem of information requirements determination by information systems personnel.

3. Transfers the information system implementation process to users. (14:429)

On the other hand, they also discuss some of the added risks involved with UDA:

1. Eliminating the analyst from the development process may also eliminate a needed outside view of the problem.

2. Lack of user knowledge may result in inconsistent standards and lower quality of systems.

3. There may be an additional risk from encouraging private information systems of also encouraging information hiding by individuals. (14:430-431)

This concludes discussion of the different approaches to information systems development. The next chapter will

focus on constructing a suitable model with which to compare specific methodologies.

IV. Building the Framework

Introduction

This section will develop an evaluative framework to compare the effectiveness of development methodologies. The first portion of the framework is a seven stage life-cycle model. The intention here is to deliberately compare the development methodologies on their relative coverage of all parts of the information systems development cycle. Several studies reviewed by this research have used this approach almost exclusively (6:9-36, 29, 40:37-62). Additionally, the use of a life-cycle model to compare methodologies is congruent with the design intentions of most comprehensive methodologies. In other words, most comprehensive methodologies seem to approach systems development in a similar way. They typically begin with a planning stage and proceed in a more or less linear fashion through stages of construction and implementation.

As an added dimension to the life-cycle model, this research also proposes that an another set of attributes which operationalize the degree to which the methodologies support the institutionalization of the information system, is a necessary addition. The rationale for these supplemental attributes is straightforward. The development and implementation of a new information system is dependent on organizational, attitudinal, social, and technical change. The more comprehensive the support that a methodology

provides for the management and institutionalization of that change the better it can facilitate effective implementation and lasting change. Thus, methodologies which provide greater degrees of support for institutionalization will have a greater degree of overall effectiveness than those which provide lesser degrees of support.

Each individual component of the framework will be discussed in this chapter. The discussion will include a description of the necessary actions required for a methodology to provide complete coverage of the individual component.

Finally, the chapter will end with a discussion of "effectiveness." Effectiveness, in this proposed framework, is defined by the degree to which a given methodology provides for development support to the full range of the information system development cycle <u>and</u> the degree to which the methodology promotes institutionalization of the necessary changes (figure 4).

Life-Cycle Coverage	+	Support For Institutionalization	=	Effectiveness

Figure 4. Model of Development Methodology Effectiveness

<u>The Life-Cycle Dilemma</u>

Many references are available in the literature to support the concept of a systems development life-cycle (14:571, 29:23, 11:85). Even so, definition of the stages

which make up the life-cycle varies considerably from one author to another. In addition to differences of opinion concerning the stages contained in the life-cycle, there is also inconsistent use of terminology. For example, while one author's use of the term "implementation" means the installation of the system into the work place (3:15), another author uses the same term to mean the "production of executable code" (40:39).

Martin's discussion of the "traditional" development life-cycle gives the following stages:

1. Requirements
2. Specifications
3. Design
4. Programming
5. Testing
6. Integration Testing
7. Deployment
8. Maintenance (30:178)

In contrast, Colter's life-cycle model is as follows:

1. Problem Definition
2. Logical Design
3. Physical Design
4. Construction
5. Integration and Testing
6. Installation
7. Evaluation (11:85)

Yet, another view of the life-cycle is given by Wasserman in his study of software development methodologies:

1. Analysis
2. Functional Specification
3. Design
4. Implementation
5. Validation
6. Evolution (40:38)

Martin asserts that the advent of computer aided analysis and design and the use of fourth generation languages by end-users will cause major changes to the development life-cycle (30:177). In Martin and Hershey's information engineering methodology, these changes are evidenced by the automation of some of the stages. For example, Martin says with the use of application generators, the program coding phase disappears, the testing and integration phase is radically shortened, and "the time taken to create applications falls from years to months with complex applications" (30:180). Martin does note, however, that the traditional concept of the development life-cycle is important for use as a guideline and to ensure that "nothing important is forgotten" (30:177).

Colter's view of the life-cycle is a bit more conservative. He suggests that the optimal design methodology "would be one that supports all of the necessary processes in the systems development cycle" (11:84). However, he states that no existing methodology fully meets the total set of life-cycle requirements due to gaps or weaknesses in coverage in certain activities (11:85).

The Seven Stage Life-Cycle Model

Clearly, the literature is in disagreement concerning the stages of the life-cycle. However, a synthesis of the views previously expressed on the life-cycle concept coupled with consideration of the approaches to systems development

presented in chapter three give rise to an acceptable life-cycle to be used for comparative purposes. This research proposes the following operationalization of a seven stage information systems development life-cycle:

Stage 1. Strategic Analysis

This phase is an important first step due to the need to align the information systems objectives with those of the organization (14:456, 5:12).

Measuring Stage 1 Success. Strategic analysis is very difficult to accomplish effectively due to the complex relationships between the organization and its environment at the strategic level. However, since we will be attempting to determine the effectiveness of development methodologies in this regard, then one could consider the detail with which the methodology covers this stage as a measure of effectiveness. Many methodologies do not consider this stage at all, while others deal with it in varying degrees of detail. A successful strategic analysis should clearly delineate the goals, objectives, and strategies of the organization and assure that information systems planning is in agreement with the strategic course of the organization (14:456).

Stage 2. Requirements analysis

The requirements analysis phase is typically concerned with understanding the problem and describing the activities involved. Davis and Olson describe it as determining the

requirements for a feasible and cost effective system (14:572).

<u>Measuring Stage 2 Success</u>. Successful requirements analysis should:

1. Assist the analyst to constrain and construct the problem space (14:479)

2. Be flexible enough to apply at all levels of the organization (14:479)

3. Involve informed users in the definition of the problem and proposed solution (14:479)

4. Be thorough enough to provide assurance that the requirements are complete and correct (14:479)

<u>Stage 3</u>. Logical Design

According to Davis and Olson (14:577), this is a user oriented design which may include the following:

- User-oriented application description. Distinguishes manual operations from automated operations performed by the application system.

- Inputs for the application with general description of each.

- Outputs produced by the application with general description of each.

- Functions to be performed by the application system.

- General flow of processing with relationships of major programs, files, inputs, and outputs.

- Outlines of operating manuals, user manuals, and training materials needed for the application.

- Audit and control processes and procedures for ensuring appropriate quality in use and operation of the application.

Measuring Stage 3 Success. As discussed by Davis and Olson, this stage could be characterized as the general design which treats the functions of the system as black boxes (14:577). This research proposes that the success of this phase should be measured by how clearly it outlines the inputs, outputs and functions to be performed by the application. In addition, success should be measured by how understandable this stage is to both the users and the analysts involved in the process.

Stage 4. Physical Design

This is a detailed design of flows and processes in the application processing system and preparation of program specifications (14:573). This phase, which is based on the logical design and the requirements analysis, provides the basis for physical database design, program development, and procedure development. It is generally the process of defining the "black boxes" described in the previous stage (14:577).

Measuring Stage 4 Success. According to Davis and Olson, successful physical design should, as a minimum include the following:

1. System design showing flow of work, programs, and user functions
2. Control design showing controls to be implemented at various points in the flow of processing
3. Hardware specifications

4. Data communications requirements and specifications

5. Overall structure of programs required by the application

6. Security and backup provisions

7. Quality assurance plan for the remainder of the development (14:578)

Stage 5. Construction

This phase includes the production of executable program code to include: physical database design, program development, and procedure development. According to Wasserman, "the code should adhere to the precepts of structured programming, with emphasis on comprehensibility of code" (40:39). Wasserman reminds that coding is only a small portion of the software development process and that it can not make up for poor analysis or design practices (40:39). Testing is assumed to be a part of each portion of the construction phase.

Measuring Stage 5 Success. This research will consider a methodology successful in this stage if it supports the coding process well. The physical design should support the concepts of modularity and structured programming aiding in a straightforward flow in logic.

Stage 6. Implementation

According to Lucas, implementation refers to the entire change process associated with a new system. He notes that computer professionals often define this stage too narrowly as a phase of systems design (28:72).

A recent study by Kwon and Zmud agrees that "most studies focus on small pieces of the MIS implementation puzzle, without considering larger issues" (27:231). The implementation stage is not just the installation and operation of the new system. Instead, this phase should be the execution of plans that were formed in the earlier stages of the life-cycle when the goals and objectives for the system were defined. It should include all preparations necessary to make the system successful. Such things as budgeting, training programs, and the allocation of resources fall in this stage. In addition, the execution of specific intervention strategies for the management of change will fall in this stage.

This stage is on-going throughout the entire system development process. The building blocks for the implementation phase are derived from the rest of the development process. Therefore it must, by its nature, be planned and executed in parallel to the other stages of the life-cycle and throughout the development process.

<u>Measuring Stage 6 Success</u>. As Lucas points out, researchers tend to measure successful implementation against some form of efficiency criterion instead of effectiveness. Nevertheless, this research agrees with Lucas that there are two plausible methods to measure the effectiveness of implementation (28:73).

In the first method, where the use of the system is voluntary and at the discretion of the user (such as summary data or a decision support system), high levels of use can be adopted as a "sign of successful implementation" (28:73). This amount of usage could be measured by interviewing users, using questionnaires, or monitoring the system.

On the other hand, where system use is mandatory, Lucas suggests "employing the user's evaluation of the system as a measure of success" (28:73). This can be accomplished through an examination of user satisfaction, measuring the timeliness and accuracy of information, or even calling upon a group of information systems experts to evaluate the design and operation of the system. As stated by Lucas:

> Favorable attitudes on the part of users should be extremely important in implementation; attitudes have an action component, and favorable attitudes are consistent with high levels of use and satisfaction with a system. (28:73).

Stage 7. Evaluation

The evaluation stage is the post audit of the system to ensure effectiveness and efficiency.

Measuring Stage 7 Success. This is an important part of the development process and should not be overlooked by the methodology. Successful support for this stage should include a planned schedule for evaluating the operation and maintenance of the system. Good post audit procedures should be specific, formalized, and well planned

in advance to ensure that the application continues to meet the needs of the organization.

This seven stage model of the system development life-cycle is operationalized for use as part of the evaluative framework of this research (see figure 5). Methodologies will be compared based on their relative support for each stage of development.

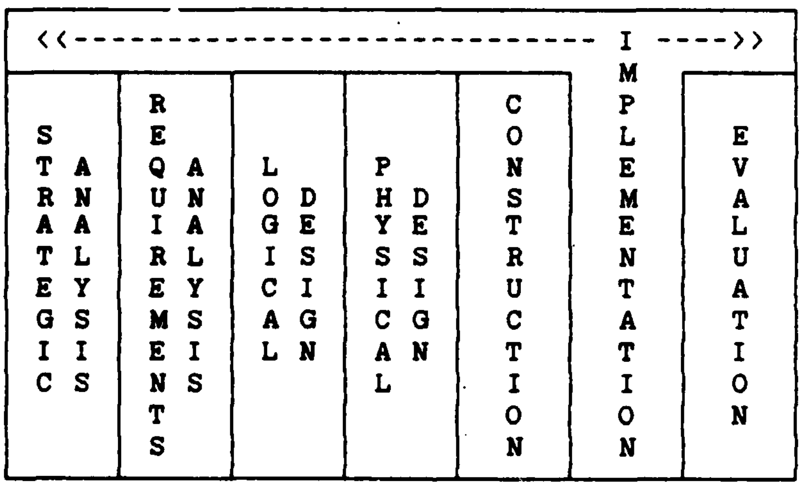


Figure 5. Seven Stage Life-Cycle Development Model

Change and Institutionalization

The average employee may never come face-to-face with the concept of the life-cycle development process. Even so, the manager and the employee are often dramatically affected by its results. In seeking attributes of development methodologies that would be important from the perspective of the manager and the employee, one must first ask what effect computerization is going to have on the organization.

Obviously, specific effects on organizations cannot be discussed without, in turn, having specific cases of applications and organizations to study. However, at a more general level, one can state with reasonable certainty that successful computerization necessarily depends on a wide range of organizational, attitudinal, social, and technical change. The change may be as simple as a secretary learning to use a word processor or a boss becoming familiar with electronic mail. On the other hand, computerization might require widespread change, such as the case in which a manufacturing plant's entire system of operation, organizational structure, social interactions, and attitude toward automation must be transformed. In any event, the very nature of computerization dictates that some degree of change is necessary to make the information system last in the organization.

Kwon and Zmud postulate that information systems implementation represents a form of diffusing technological innovation throughout the organization (27:231). They promote a comprehensive model of the implementation process which merges two major streams of research in this area: organizational innovation and information systems implementation (27:244).

The model of effectiveness depicted in figure 4 (page 43) incorporates the perspective of Kwon and Zmud by proposing that a comprehensive system development method-

ology is effective only if it acts to help institutionalize change. Many traditional methodologies have tended to focus on a narrow technical solution to systems development. They have tended to ignore many of the more broad organizational, attitudinal, and social aspects of development that accompany successful and lasting implementation. It is the contention of this research that the overall effectiveness of any methodology can be judged by the degree of coverage it provides of both the life-cycle development process and the process of institutionalization.

<u>Institutionalizing Change</u>. Goodman and Dean (20:285-291) assert that the significance of institutionalizing change should be apparent:

> If one is interested in bringing about long-term changes in productivity and in the quality of working life, labor-management relationships, and organizational effectiveness, then we must know more about why some change programs remain viable while others decline. (20:285)

A study conducted by Goodman and Dean found that only one third of the change programs that had been successfully implemented "exhibited some reasonable level of persistence" (20:285). They point out that these low rates of persistence pose a very practical problem for organizational management "given the huge amounts of human and financial resources allocated to programs of change" (20:285).

They define institutionalization as:

> A behavior that is performed by two or more individuals, persists over time, and exists as a

part of the daily functioning of the organization. (20:286)

Institutionalization, according to Goodman and Dean, is a function of the following actions (20:289):

1. Plan for institutionalization. Be sure that resources are aimed at long-term maintenance of the program as well as initiating it.

2. Be aware of congruence problems. The more different the changes are from the norms and values of the organization, the more difficult it will be to make the changes persist.

3. State specific program goals. The more specific and concrete the objectives of the program the better.

4. Formal procedures. Formal procedures to implement the change increase the degree of institutionalization.

5. Limited, short-term use of consultants. Programs should be instituted in such a way that the organization learns to handle the change without the long-term need for consultants.

6. Participation. High levels of commitment arise from voluntary participation in the programmed change.

7. Training over time. Training must be redone periodically to reinforce the change.

8. Diffusion. Institutionalization is enhanced by spreading the change over as wide an area in the organization as possible.

9. Evaluation. An accurate feedback mechanism is necessary in order to assess the validity of the program and make adjustments so that it can adjust, grow, and remain viable over time.

These factors can be operationalized as attributes on which to compare systems development methodologies. The comparison here would essentially be one judging the

relative likelihood of the information system derived from different methodologies becoming institutionalized.

Transformation into Systems Development Attributes. Based on Goodman and Dean's research, the following attributes of development methodologies should be instrumental in obtaining a high degree of institutionalization for any proposed development process.

Plan for institutionalization. This attribute correlates well with effective strategic level analysis. Here, the analysis should ensure resources are aimed at long-term objectives and goals of the organization as well as long-term maintenance of the program.

Awareness of congruence problems. This is operationalized by an assessment of the degree of the organization's underlying readiness for change (e.g. computer literacy of the organization) in comparison to the degree of change the new system will impose on the existing organization. Understanding the degree of change involved and capabilities to meet those changes is an important first step in deciding what actions should be taken in the management of the proposed change.

Statement of specific program goals. This attribute translates into the need to plan the implementation of the new system from the ground up. Systems developers should ensure that appropriate individuals close

to actual applications are involved and that they understand the specific goals and objectives of the plan.

Formal procedures. From the goals and objectives, specific activities and milestones should be established for implementing the new system.

Limited, short-term use of consultants. The system implementation plan should be such that individuals within the organization are empowered by experts to handle the fine tuning that will be necessary to adapt the system and organization over the long-run.

Participation. Systems developers should ensure that user participation is actually carried out in both the initial and the later stages of the change. Users may need training in order to participate meaningfully and productively. User participation must be perceived as genuine to facilitate the "buy-in" by the participants.

Training over time. During the evaluation stage, variances may move outside acceptable tolerances. On-going training programs must be available to reinforce the change and keep the program on track with organizational needs.

Diffusion. Ensure that all areas in the organization that could benefit from the change are included. The wider the change is diffused throughout the organization, the better chance it has of becoming institutionalized.

Evaluation. An accurate feedback mechanism is necessary in order to assess the validity of the program and

make adjustments so that it can adjust, grow, and remain viable over time.

In summary, the framework developed to compare the effectiveness of information systems development methodologies takes into account both completeness of coverage of the development life-cycle (figure 5) and attributes which contribute to the institutionalization of change and innovation (figure 6). Support for institutionalization is presented as a necessary addition to the life-cycle model in order to judge development methodologies on their ability to facilitate effective implementation and lasting change.

INSTITUTIONALIZATION FACTORS							
PLANNING FOR	CONGRUENCE AWARENESS	SPECIFICITY	FORMAL PROCEDURES	LIMITED CONSULTANTS	PLANNED USER PARTICIPATION	TRAINING OVER TIME	DIFFUSION

Figure 6. Institutionalization Factors Promoting Lasting Change

The Concept of Effectiveness

As Davis and Olson point out, it is often difficult to distinguish between effectiveness and efficiency. However, in their discussion of effectiveness versus efficiency, they note that effectiveness is output oriented while efficiency is process oriented (14:287). More specifically stated:

> Effectiveness is a measure of 'goodness' of output, while efficiency is a measure of the resources required to achieve the output. (14:287)

As shown in figure 7, this research proposes that the effectiveness of a methodology is equal to the degree of coverage of the two main components of this model: the systems development life-cycle and the process of institutionalization. The next chapter will present comparisons of selected methodologies based on this framework.

Life-Cycle Coverage	+	Support For Institutionalization	=	Effectiveness

Figure 7. Model of Development Methodology Effectiveness

V. Comparison of Methodologies

Introduction

This chapter provides comparisons of the methodologies discussed in this research based on the previously developed framework. Coverage of the framework's individual attributes will be qualitatively and subjectively evaluated from their descriptions in the literature. The scales are as follows:

Life-Cycle Coverage

(1) No coverage

(2) Medium coverage

(3) Good coverage

Support for Institutionalization

(1) No support

(2) Medium support

(3) Good support

Following the verbal description evaluating the methodologies against the framework, the results of the discussion are presented in a matrix format which is similar to the graphic representation used by Colter in his study of analysis techniques (12:56-66).

Strategy Set Transformation

Life-Cycle Coverage. This methodology is the only one found which aimed specifically at strategic level analysis. It is rated good in its coverage of this area. It does not

appear applicable to other aspects of the life-cycle.

Support for Institutionalization. SST is rated medium in its coverage of both planning for institutionalization and planned user participation. The methodology is not specifically aimed at the institutionalization process. However, it appears to be an effective means of identifying the organizations goals and objectives. In addition, the medium rating in the planned user participation category results from its stated need for validation with managers and users.

Critical Success Factors

Life-Cycle Coverage. This methodology is aimed primarily at the requirements analysis stage. It is rated good in its coverage of this stage based on its ability to:

(1) assist the analyst in bounding the problem space

(2) be flexible enough to apply at all levels of the organization

(3) involve informed users in the definition of the problem and proposed solution

It is not applicable to other aspects of the life-cycle.

Support for Institutionalization. Critical Success Factor analysis is rated medium on specificity (its ability to define organization and program goals). It rates medium in coverage of planned user participation but it must be remembered that this user participation is limited to the

requirements analysis stage. In addition, in accordance with the previous discussion of the methodology, it is rated medium on its ability to diffuse that information throughout the organization as it has the capability to involve all levels of employees in the determination of critical success factors.

Business Systems Planning

Life-Cycle Coverage. BSP rates good in the area of strategic planning. It does consider the organization's business wide perspective including its environment. BSP is also rated good in its coverage of requirements analysis, logical design, physical design, and construction. It is rated good in the implementation stage because it considers this process from both the top down and bottom up and throughout the design process. Further, no other structured methodologies found by this research were as complete and thorough in the planning and preparation for implementation as BSP.

Support for Institutionalization. BSP rates good in the area of planning, specificity, and formal procedures due to its extremely comprehensive nature. However, it must be noted that the view of the planning process is purposefully viewed from the top management perspective. This perspective may not always give the organization the most realistic view of the effects of implementing a new system. BSP is rated medium on planned user participation. Even though IBM

has extensive support for training and support, the issues of training over time, diffusion, and evaluation are not specifically addressed by the literature reviewed by this research.

Structured Analysis and Design Technique (SADT)

Life-Cycle Coverage. The life-cycle coverage of SADT is described by Wasserman (40:37-43). Wasserman's study rates SADT good in coverage of requirements analysis, logical design and physical design. However, this research feels that his ratings relative to requirements analysis are based more on SADT's technical ability to handle a given set of specifications rather than its ability to elicit the requirements from managers and users of the system. It appears that, to be more effective during this stage, SADT could be coupled with another methodology such as CSF or BIAIT. Therefore SADT is rated medium in its coverage of requirements analysis. In agreement with Wasserman, it is rated good in the areas of logical and physical design.

Support for Institutionalization. No support for institutionalization factors is noted.

Active and Passive Component Modeling (ACM/PCM)

Life-Cycle Coverage. Again, Wasserman's study is helpful in determining the life-cycle coverage of this methodology (40:39-43). As with SADT, it is also rated medium in the area of information systems requirements

determination even though Wasserman rated its coverage good. Again, this is due to its technical orientation as was the case with SADT. This research accepts Wasserman's ratings of good coverage of the stages of logical design, physical design, and construction.

Support for Institutionalization. There was no evidence found that ACM/PCM provides coverage in this area.

Business Information Analysis and Integration Technique (BIAIT)

Life-Cycle Coverage. BIAIT's coverage is rated good in both the areas of requirements analysis and logical design. The essence of BIAIT is described by Carlson as a process designed for full agreement between the end-user and the analyst prior to action (9:220). In this sense, a fairly comprehensive logical model of the organization's information requirements is developed.

Support for Institutionalization. It might appear on the surface, since BIAIT consists of an interview technique, that it would be useful in determining congruence problems. However, it is clear from the description of the methodology, that it is not focused in that area. Instead, it is aimed at a technical solution to the business problem. BIAIT is rated medium on specificity due to its rigid nature of defining the problem. Even so, there is no evidence to suggest that it is intended to assist in the management of the change process. Further, BIAIT is rated medium in the

area of user participation because it is an interview technique.

Mumford's ETHICS

Life-Cycle Coverage. ETHICS is rated based on its description published by Hirschheim (23:111-118). It is rated good in the areas of strategic analysis, requirements analysis and logical design. In strategic analysis, ETHICS clearly addresses the issues of system boundaries, environmental relationships and future possibilities. Its coverage of the requirements analysis and logical design of the information system is comprehensive through the socio-technical process involved.

In addition, it is rated medium in the area of implementation. Even though it provides no support for physical design and construction, it provides good planning for the implementation of the system from the very beginning.

Support for Institutionalization. ETHICS is rated good in planning for institutionalization due to its thorough analysis of both the social and the technical aspects involved in the implementation of an information system. It is rated good for awareness of congruence problems because of its identification of both technical and social constraints. ETHICS is also rated good for specificity of the planning and implementation process.

ETHICS is most definitely rated good in the area of planned user participation for that is the essence of the socio-technical perspective. No coverage is noted for other attributes.

Pava's Socio-technical Design Methodology

Life-Cycle Coverage. Pava's methodology is also being rated based on its description by Hirschheim (23:125-129). It is rated good in the area of strategic analysis. Clearly, the analysis of the global mission, philosophy, and the key internal and external factors influencing the organization are an essential part of this methodology (23:127). Additionally, this methodology is rated good in the area of requirements analysis and logical design. Through its socio-technical attack of the problem, it attempts to define a model of the organization which includes both technical and organizational procedures required.

Finally, it is also given a medium rating in the area of implementation based on the comprehensive socio-technical approach to the design process which keeps the users heavily involved.

Support for Institutionalization. This socio-technical methodology also rates well in institutionalization factor coverage. It rates good in the area of planning due to its in-depth consideration of the objectives and strategies of the organization along with the needs of its people. It is

also rated good in the area of congruence awareness. The methodology specifically addresses the identification of divergent values which could hinder the project. The area of planned user participation is also rated good due to the socio-technical nature of the methodology.

Information Engineering (IE)

Life-Cycle Coverage. IE is rated good in both strategic and requirements analysis. The methodology specifically addresses the linking of information systems requirements to top management's strategic planning. It makes use of a critical success factor analysis for the requirements analysis function. Further, IE is rated good in its coverage of logical design, physical design, and construction. Through the automation of these three processes, there is a natural linkage and flow from one process to another, resulting in an application which remains congruent with previous stages in the life-cycle.

Finally, IE is rated good in the area of implementation. The methodology considers the use of prototyping and end-user development important to the development process. It also plans for data modeling (of the data base) to ensure that information will be compatible to cross functional boundaries. No coverage is noted of an evaluation stage.

Support for Institutionalization. The comprehensiveness of the methodology compels a good rating in the

area of planning for institutionalization. It rates good in the areas of specificity and limited short term use of consultants. The automation of the design process makes the maintenance and evolution of the system manageable by the business without the need for consultants in the long-run.

Planned user participation is rated good as well. IE is rated medium for diffusion (based on its comprehensive nature) even though this category is not specifically addressed. No other coverage was apparent.

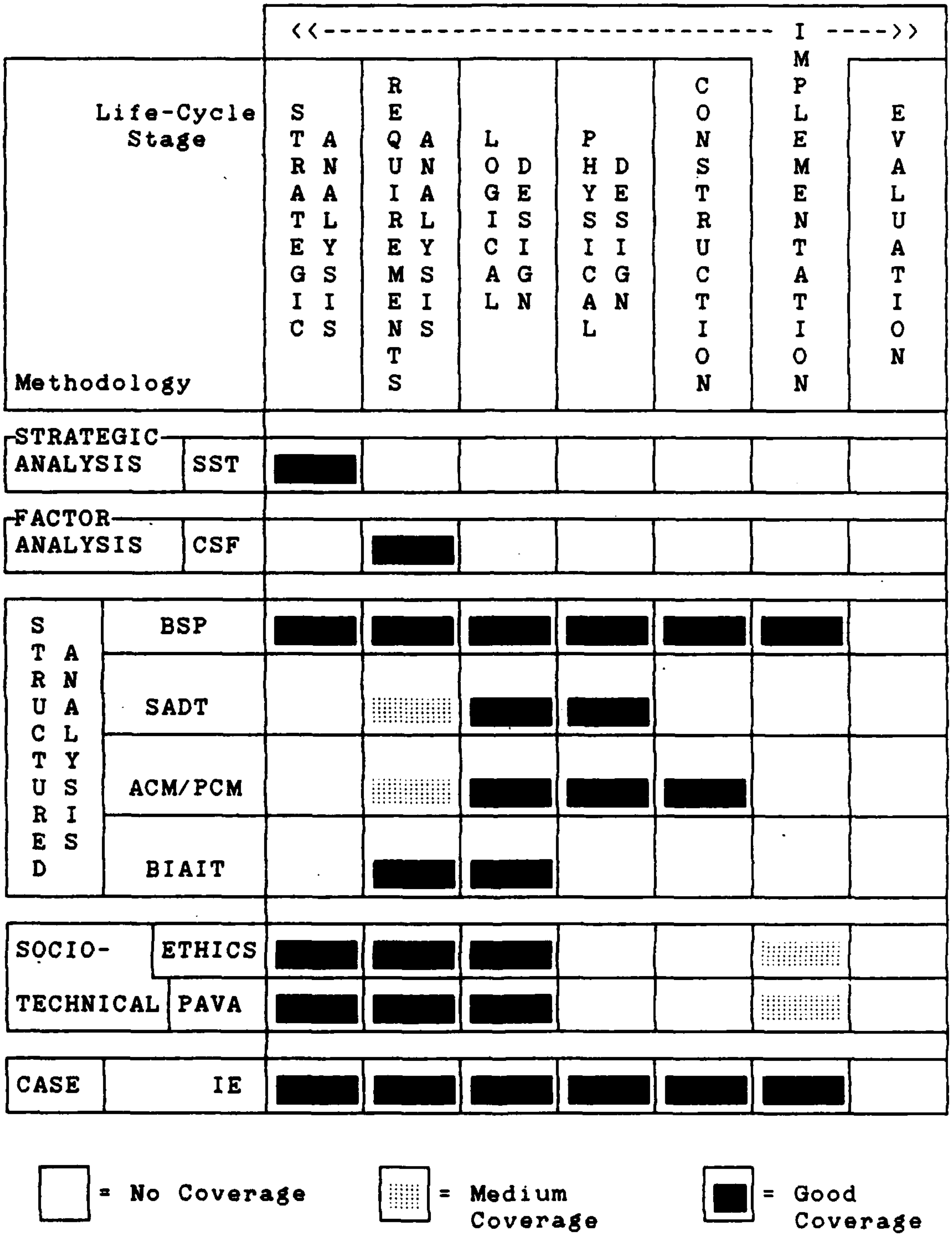


Figure 8. Life-Cycle Coverage of Selected Methodologies

Methodology		INSTITUTIONALIZATION FACTORS: PLANNING FOR	CONGRUENCE AWARENESS	SPECIFICITY	FORMAL PROCEDURES	LIMITED CONSULTANTS	PLANNED USER PARTICIPATION	TRAINING OVERTIME	DIFFUSION
STRATEGIC ANALYSIS	SST	Medium					Medium		
FACTOR ANALYSIS	CSF			Medium			Medium		Medium
STRUCTURED ANALYSIS	BSP	Good		Good	Good		Medium		
STRUCTURED ANALYSIS	SADT								
STRUCTURED ANALYSIS	ACM/PCM								
STRUCTURED ANALYSIS	BIAIT			Medium			Medium		
SOCIO-TECHNICAL	ETHICS	Good	Good	Good			Good		
SOCIO-TECHNICAL	PAVA	Good	Good				Good		
CASE	IE	Good		Good		Good	Good		Medium

☐ = No Support ▒ = Medium Support ■ = Good Support

Figure 9. Support for Institutionalization by Selected Methodologies

VI. Conclusions and Recommendations for Future Research

This evaluation of development methodologies leads to several conclusions. First, the traditional focus of structured methodologies has been too narrowly confined to the technical aspects of systems development. Second, the socio-technical methodologies reviewed are unable to stand alone as methodologies for developing information systems. Third, the merging of CASE methodologies with the socio-technical approach could be a very effective way to break out of the traditionally narrow focus of a technical solution to systems development and implementation.

Conclusion One: Structured Focus Too Narrow

Structured methodologies (except for BSP) have tended toward a narrow focus in support of logical and physical design. They should be expanded into a broader framework which would include proactive management of the change process which is inevitable with the implementation of innovation.

Even though BSP gives the appearance of being extremely comprehensive, the fact that it requires voluminous documentation and the involvement of many managers in the organization for extended periods of time can make it difficult to use. Moreover, this research finds little evidence that BSP makes a deliberate attempt to manage the

change process. Instead, its focus seems much more intent on defining the processes of the organization.

Conclusion Two: STS Approach does not Stand Alone

Socio-technical methodologies pay more attention to factors which provide both a complete requirements analysis and the institutionalization of the change process. However, these methodologies provide few specifics which are easily translated into program code. Therefore, in order for the analyst to effectively use the socio-technical approach, the results of the analysis stage must be further refined using some form of structured analysis technique (such as SADT) to complete the physical design. In other words, the socio-technical methodologies reviewed in this research do not appear capable of developing an information system on their own. They appear to be organizational and management oriented. They are less specifically concerned with the more rigorous details of system design and program coding.

Conclusion Three: A Merger Might Provide the Solution

Of all the classes of methodologies examined by this research, the CASE methodologies appear to be the most promising in their ability to allow development efforts to focus on the actual problem at hand vice the complex aspects of the solution to the problem. One implication of this conclusion is that CASE methodologies, if merged with

concepts of innovation management (e.g. socio-technical philosophies, institutionalization factors), could provide the means by which systems developers can pay more attention to the broader aspects of design and implementation. These broader aspects might include such items as job satisfaction, task identity or task variety.

The use of prototyping and fourth-generation languages would fit well with socio-technical ideals. Prototyping naturally facilitates genuine user participation in the design process. Further, fourth-generation languages could provide the needed flexibility to allow changes in design to keep pace with the constantly evolving demands of users.

Recommendations for Future Research

This research was not intended to be focused on testing specific hypotheses. Rather, this research effort can best be characterized as descriptive, exploratory, and hypothesis generating. In this regard, a number of hypotheses are derived from this study for future research.

Hypothesis 1: Development efforts which pay attention to implementation (the management of the change process) throughout the development life-cycle will be more effective than those which view the implementation process as merely a sequential phase in the development process.

Hypothesis 2: A development methodology which pays conscious attention to planning for institutionalization

will produce more effective and longer lasting information systems than one which does not.

Hypothesis 3: Development efforts which measure congruence awareness and use this data to outline specific goals for planned user participation will meet less resistance in the installation and operation of the system. This will lead to greater effectiveness.

Hypothesis 4: Development efforts which have specific program goals and objectives, and formal procedures to accomplish them, will have a greater chance of long lasting success than those which do not.

Hypothesis 5: New systems implementations which are intentionally diffused into the widest spectrum of the organization possible will have a greater chance of institutionalization than those which are not.

Hypothesis 6: Development methodologies which provide for specific evaluation criteria, with plans to modify training over time, will facilitate the maintenance of effective systems better than those which do not.

Testing the Hypotheses

It is recognized that tests for the above six hypotheses will be difficult to devise. However, recalling the previous discussions concerning congruence awareness and effectiveness versus efficiency, it is certainly possible to formulate tests which could measure dependent variables such as system completeness, user satisfaction, and institution-

alization. Operationalization of these variables could come from a combination of questionnaires, observation, and interviews. One reliable test of system effectiveness could be to measure the amount of longitudinal system usage. Lucas suggests that testing the subjective timeliness and accuracy of information could be an additional measure of system effectiveness (28:73).

The difficulty in testing the above hypotheses undoubtedly lies in the fact that gathering the type of data required (to make factual conclusions as to the long-term effectiveness of development efforts) will necessitate a longitudinal study with the cooperation of many people and organizations throughout the process.

Informal "Non-Traditional" Conclusions

Now that considerable effort has been spent discussing the subject of information systems and development methodologies in terms of the current literature from an academic perspective, I'd like to deviate from the norm. I'm now going to break tradition and speak directly to you, the reader, and give you in my own words, a synthesis of what I feel are currently the most important aspects of information systems development methodologies from the entire body of literature reviewed in this research.

To do this, let's pretend that I'm now a manager whose organization is about to undergo a transformation involving the installation of a new computer system. As I see it,

there are three key points to keep in mind as the new system is developed and the implementation of the system unfolds. These key points are careful planning, planned user participation, and evaluation and training over time.

Key Point One: Careful Planning. First and foremost on my list of important things to consider would be the concept of careful, structured planning. To be truthful, it seems that this is probably one of the most talked about issues in the literature. However, it still receives inadequate attention in practice. This is probably because planning requires effortful thinking. Nevertheless, it is essential that a detailed plan be established which outlines the specific and formal procedures to be followed.

Additionally, I would ensure that my planning effort began with a strategic level analysis of the organization. This analysis would pinpoint, as accurately as possible, the long-term goals and objectives of the organization. Once strategic direction is identified, one can proceed to evaluate the pros and cons of adding an information system to the organization. It might very well be that the long-term objectives identified don't lend themselves to computerization. Conversely, if the strategic analysis looks favorable we can proceed with some degree of consensus and direction on systems development.

As discussed in the previous chapter, one of the most overlooked areas of the systems development process is in

planning the implementation of the system with institutionalization in mind. The implications of poor implementation planning are obvious. The best of systems with poor implementation and institutionalization planning may neither be fully developed or retained in the organization.

Key Point Two: User Participation. It is absolutely clear that user participation is a key to developing effective information systems. There are at least two reasons for this. First, it is generally agreed upon by organizational development professionals that user participation is the key factor capable of reducing resistance to change. In turn, resistance to change is one of the most common causes of implementation failures. Second, through prototyping, user participation has the capability to greatly improve our ability to develop a useful system.

Users often don't know what they want in an information system until after they've tried using some approximation of the system. Experimentation helps users to realize system capabilities and weaknesses. Even in the most routine and simple development efforts, the concept of prototyping appears capable of saving enormous amounts of time and money. Further, prototyping provides for appropriate involvement of the user in an setting where his/her suggestions can be quickly incorporated into the development process. This type of "real" user involvement has the

potential for increasing system usability. So, I would insist on having user participation in the form of prototyping as a part of my system development effort if at all possible.

<u>Key Point Three: Evaluation and Training Over Time</u>. Finally, in developing my information system, I would ensure that formal objectives were defined for evaluating the effectiveness of the system after it was installed and operational. It seems appropriate to find both objective and subjective means of measuring changes in productivity. Both quality and volume of office output prior to the new system could be measured and then compared to future measures. I must note, however, that a one-time evaluation of the system is not effective. In order to ensure that the system continues to meet the needs of the organization over time, a regular and periodic evaluation of effectiveness is in order. If, after an evaluation has taken place, it is determined that the system is deficient, then it will be important to again involve users in the modification process and to amend training programs appropriately.

Appendix: Additional Methodology Descriptions

Structured Analysis

Structured Analysis and Design Technique (SADT). According to Lucas, the objective of SADT is to "force structure on the unstructured systems analysis and design task (28:140). This technique was developed by a private firm and consists of:

1. A graphic language for building models
2. A method for developing models
3. Management practices for controlling the development (28:140)

SADT guides the analyst into a top-down structured decomposition of the problem with the help of a graphic modeling language (28:140). Activity and data diagrams are the main components of the methodology using boxes to show activities and lines to connect the boxes to show data interface between them (28:140).

Discussion. Ross notes that the methodology "permits teams of people to work and interact as one mind attacking complex problems" (38:161). However, this research notes that the language itself and the diagrams it creates can be quite difficult for users to comprehend. In addition, understanding the unique mode of looking at a problem through the eyes of this methodology can be somewhat difficult (even for the experienced analyst who has been used to data flow diagrams or some other form of functional

decomposition). It does seem, however, that the methodology is able to create a detailed description of the problem which flows smoothly into the coding process.

Business Information Analysis and Integration Technique (BIAIT). This methodology focuses on the need to get full agreement between the user-managers and the analyst prior to anyone writing code or installing a manual system (9:220). It consists of a technique made up of seven questions which bound the problem space. These questions are posed to problem relevant personnel in an iterative fashion. BIAIT is made up of four stages:

1. Create a generic model of the organization.
2. Customize the model by interviewing decision makers in the organization to see how closely the model fits with their perceptions.
3. Prioritization and value analysis to decide what applications are most important to top management.
4. Convert specifications to a running application. (9:220-221)

Discussion. BIAIT is described by Carlson as a simple analytical tool. It can serve as a useful communication device between the user and the analyst (9:222). However, this research must note that it is a fairly limited tool which focuses narrowly on the requirements analysis spectrum of the development life-cycle.

Active and Passive Component Modeling (ACM/PCM). This methodology was developed mainly in a university environment

with some large system successes (6:14). According to Brandt, ACM/PCM begins with the modeling of data and transactions and ends up with specifications that are close to program level. Brodie describes the methodology as useful in designing large size 'database-intensive' applications (7:41). The methodology claims to cover the complete life-cycle process of development from requirements formulation and analysis to evolution of the system. However, the only detailed discussion found of the methodology involved discussion of the logical design and specification steps

Discussion. Some of the weak points Falkenberg et al. describe are:

1. Too complicated from a user point of view.

2. Main objective of a 'precise abstract model' unattained.

However, they also include some of the following comments as strong points of the methodology:

1. Good blend of procedure and object orientation.

2. Structural and behavioral properties well integrated.

3. Modular behavioral model (19:172)

According to Maddison and others:

> The methodology has potential although to be readily usable the BETA Language ought to be made more user friendly. Also, without additional information both on the remaining phases of physical design, implementation, and on the interface between phases it is difficult to assess practical usefulness. (29:19)

Socio-Technical Systems Analysis

Pava's Socio-technical Design Methodology. The following six steps form the pivotal characteristics of the methodology:

1. Mapping the target system by tracing the sequence of deliberations. Deliberations are the reflective and communicative behaviors concerning a particular topic.

2. Structuring for maximum self design. This stage involves (a) gaining access to senior people whose support is essential for success; (b) obtaining formal approval from senior management; (c) establishment of a design group made up of key departmental members along with a person to act as a facilitator.

3. Initial scan. In this stage design group members need to do a strategic analysis identifying the goals of the organization and their unit. They also need to identify organizational philosophy on the management of its people and key internal and external factors involved.

4. Technical analysis. This is an iterative process to analyze the technical subsystem of the unit. It involves an examination of the tools and procedures involved in converting inputs to outputs.

5. Social analysis. This stage involves an analysis of the social subsystem which uses the technical subsystem to convert inputs to outputs. According to Hirschheim, its main task is to identify divergent values, interdependent parties, role networks, and discretionary coalitions.

6. Work system design. In this stage, an attempt is made to identify the best fit between the technical and social subsystems. The objective, as reported by Hirschheim, is to "create a variety-increasing work system which embraces the notion of 'redundant functions'" (23:128). The concept of redundant functions refers to the basic ideas of semi-autonomous work group design common in the sociotechnical

> literature. The basic idea is that "more than one person possesses any one skill and each person possesses more than one skill" (23:128-129).

<u>Discussion</u>. According to Hirschheim, Pava's methodology is similar to that of ETHICS but extends socio-technical design more into the domain of the office (23:125). One key to this methodology is the fact that the office needs to be viewed as an open system (this is a fairly straightforward concept as it relates to factory work, but viewing office work in this manner is a bit more abstract) (23:125).

Bibliography

1. Alavi, M. 'The Evolution of Information System Development approach: Some Field Observations,' Database, 14(3): 19-24 (Spring 1984).

2. Benjamin, Robert L. 'Information Technology in the 1990s: A long Range Planning Scenario,' MIS Quarterly, 6(2): 11-32 (June 1982).

3. Bostrom, Robert P. and J. Stephen Heinen. 'MIS Problems and Failures: A Socio-Technical Perspective - Part I: The Causes,' MIS Quarterly, 1(3): 17-32 (September 1977).

4. -----. 'MIS Problems and Failures: A Socio-Technical Perspective - Part II: The Application of Socio-Technical Theory,' MIS Quarterly, 1(4): 11-28 (December 1977).

5. Bowman, Brent J., Gordon B. Davis and James C. Wetherbe. 'Three Stage Model of MIS Planning,' Information and Management, 6(1): 11-25 (February 1983).

6. Brandt, Ingunn. 'A Comparative Study of Information Systems Design Methodologies,' Information Systems Design Methodologies: A Feature Analysis, 9-36, edited by T. W. Olle and others. Amsterdam: North-Holland, 1983.

7. Brodie, Michael L. and Erico Silva. 'Active and Passive Component Modeling: ACM/PCM,' Information Systems Design Methodologies: A Comparative Review, 41-91, edited by T. W. Olle and others. Amsterdam North-Holland, 1982.

8. Carey, Elizabeth M. and Dr. Philip K. Y. Young. 'Incorporating the PC into the Workplace,' Journal of Systems Management, 36(4): 34-37 (April 1985).

9. Carlson, Walter M. 'Business Information Analysis and Integration Technique (BIAIT) - the New Horizon,' Advanced System Development/Feasibility Techniques, 213-222, edited by J. Daniel Cougar and others. New York: John Wiley and Sons, Inc., 1982.

10. Cerveny, Robert P. et al. "The Application of Prototyping to Systems Development: A Rationale and Model," Journal of Management Information Systems, 3(2): 52-62 (Fall 1986).

11. Colter, Mel A. "Evolution of the Structured Methodologies," Advanced System Development/Feasibility Techniques, 73-96, edited by J. Daniel Cougar and others. New York: John Wiley and Sons, Inc., 1982.

12. -----. "A Comparative Examination of Systems Analysis Techniques," MIS Quarterly, 8(1): 51-66 (March 1984).

13. Cougar, Daniel J., Mel A. Colter and Robert W. Knapp. Advanced System Development/Feasibility Techniques. New York: John Wiley and Sons, Inc., 1982.

14. Davis, Gordon B. and Margrethe H. Olson. Management Information Systems Conceptual Foundations, Structure, and Development. New York: McGraw-Hill Book Co., 1985.

15. Davis, William S. Tools and Techniques for Structured Systems Analysis and Design. Reading: Addison-Wesley, 1983.

16. De, Prabuddha and Arun Sen. "A New Methodology for Database Requirements Analysis," MIS Quarterly, 8(3): 179-193 (September 1984).

17. Dickson, Gary W., Robert L. Leitheiser and James C. Wetherbe. "Key Information Systems Issues for the 1980's," MIS Quarterly, 8(3): 135-160 (September 1984).

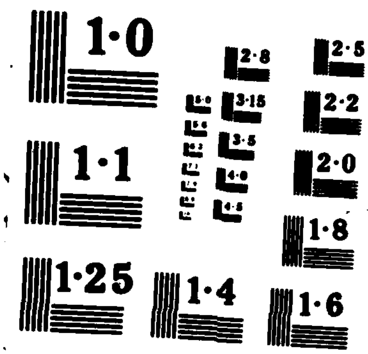
18. Ein-Dor, Phillip and Eli Segev. "Strategic [illegible] for Management Information Systems," Management Science, 24(15): 1631-1641 (November 1978

19. Falkenberg, E. and others. "Feature Ana[illegible] ACM/PCM, CIAM, ISAC and NIAM," Informat[illegible] Design Methodologies A Feature Analysis [illegible] edited by T. W. Olle and others Ams[illegible] North-Holland, 1983.

20. Goodman, Paul S. and James W [illegible] Productivity Efforts Fai[illegible] 285-291, edited by W [illegible] Business Publications [illegible]

AD-A188 957 A COMPARISON OF INFORMATION SYSTEM DEVELOPMENT METHODOLOGIES(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF SYSTEMS AND LOGISTICS 2/2

UNCLASSIFIED S D BRANCH DEC 87 AFIT/GIR/LSR/87D-1 F/G 5/2 NL


1·0
2·8
2·5
3·15
2·2
3·5
1·1
2·0
4·0
4·5
1·8
1·25
1·4
1·6

21. Gorry, Anthony G. and Michael S. Scott-Morton. "A Framework for Management Information Systems," Sloan Management Review, 55-70 (Fall 1971).

22. Guimaraes, Tor and Vasudevan Ramanujam. "Personal Computing Trends and Problems: An Empirical Study," MIS Quarterly, 10(2): 179-188 (June 1986).

23. Hirschheim, R. A. Office Automation - A Social and Organizational Perspective. New York: John Wiley and Sons, Inc., 1985.

24. IBM. "Business Systems Planning," Advanced System Development/Feasibility Techniques, 236-314, edited by J. Daniel Cougar and others. New York: John Wiley and Sons, Inc., 1982.

25. Konsynski, Benn R. "Advances in Information System Design," Journal of Management Information Systems, 1(3): 5-32 (Winter 1984-85).

26. Kottemann Jeffrey E. and Benn R. Konsynski. "Information Systems Planning and Development: Strategic Postures and Methodologies," Journal of Management Information Systems, 1(2): 45-63 (Fall 1984).

27. Kwon, Tae H. and Robert W. Zmud. "Unifying the Fragmented Models of Information Systems Implementation," Critical Issues in Information Systems Research, edited by R. J. Boland Jr. and R. A. Hirschheim, New York: John Wiley and Sons, 1987.

28. Lucas, Henry C. The Analysis, Design, and Implementation of Information Systems. New York: McGraw-Hill, 1984.

29. Maddison, R. N. (Ed). Information System Methodologies. Great Brittain: Wiley Heyden Ltd., 1983.

30. Martin, James. An Information Systems Manifesto. Englewood Cliffs: Prentice-Hall Inc., 1984.

31. Martin, James and E. A. Hershey III. Information Engineering, A Management White Paper. Product Brochure. KnowledgeWare, Inc., 2006 Hogback Road, Ann Arbor MI 48105, 1986.

32. Naumann, Justus D. and A. Milton Jenkins. "Prototyping: The New Paradigm for Systems Development," MIS Quarterly, 6(3): 29-44 (September 1982).

33. Nolan R. L. and James C. Wetherbe. "Toward a Comprehensive Framework for MIS Research," MIS Quarterly, 4(2): 1-19 (June 1980).

34. Paddock, Charles E. "A Critical View of Factors Affecting Successful Application of Normative and Socio-Technical Systems Development Approaches," Information and Management, 10(1): 49-57 (January 1986).

35. Rivard, Suzanne and Sid L. Huff. "An Empirical Study of Users as Application Developers," Information and Management, 8: 89-102 (February 1985).

36. Rockart, J. F. "Chief Executives Define Their Own Data Needs," Harvard Business Review, 81-93 (March/April 1979).

37. Rockart, John F., Leslie Ball and Christine V. Bullen. "Future Role of the Information Systems Executive," MIS Quarterly, Special Issue: 1-14 (December 1982).

38. Ross, Douglas T. "Structured Analysis (SA): A Language For Communicating Ideas," Advanced System Development/Feasibility Techniques, 135-163, edited by J. Daniel Cougar and others. New York: John Wiley and Sons, Inc., 1982.

39. Shank, Michael E., Andrew C. Boynton and Robert W. Zmud. "Critical Success Factor Analysis as a Methodology for MIS Planning," MIS Quarterly, 9(2): 121-129 (June 1985).

40. Wasserman, Anthony I. and others. "Characteristics of Software Development Methodologies," Information Systems Design Methodologies: A Feature Analysis, 37-62, edited by T. W. Olle and others. Amsterdam: North-Holland, 1983.

VITA

Captain Steven D. Branch was born on 10 May 1950 in Oklahoma City, Oklahoma. He graduated from Classen High School in Oklahoma City in 1968. After enlisting in the United States Air Force in June 1971, he attended the Defense Language Institute, Monterey, California, where he studied Chinese-Mandarin. His first tour of duty was as a Chinese linguist in Taiwan. He received a Bachelor of Business Administration from Central State University, Edmond, Oklahoma, in May 1977. Captain Branch was commissioned through Officer's Training School in February, 1979. He has served in various management positions in the administrative officer career field since his commissioning. These positions include administrative officer and headquarters section commander at base level; executive officer at wing level; and protocol officer and plans/program analyst at intermediate headquarters level. Just prior to entering the School of Systems and Logistics, Air Force Institute of Technology, in June 1986, he was serving as the Team Chief, Standardization and Evaluation Division, Headquarters Sheppard Technical Training Center, Sheppard Air Force Base, Wichita Falls, Texas.

Permanent address: 4720 Mulberry
Woodward OK 73801

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED

1b. RESTRICTIVE MARKINGS:

2a. SECURITY CLASSIFICATION AUTHORITY:

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE:

3. DISTRIBUTION / AVAILABILITY OF REPORT: Approved for public release; distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S): AFIT/GIR/LSR/87D-1

5. MONITORING ORGANIZATION REPORT NUMBER(S):

6a. NAME OF PERFORMING ORGANIZATION: School of Systems and Logistics

6b. OFFICE SYMBOL *(If applicable)*: AFIT/GIR

7a. NAME OF MONITORING ORGANIZATION:

6c. ADDRESS *(City, State, and ZIP Code)*:
Air Force Institute of Technology
Wright-Patterson AFB OH 45433-6583

7b. ADDRESS *(City, State, and ZIP Code)*:

8a. NAME OF FUNDING / SPONSORING ORGANIZATION:

8b. OFFICE SYMBOL *(If applicable)*:

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER:

8c. ADDRESS *(City, State, and ZIP Code)*:

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO	WORK UNIT ACCESSION NO.

11. TITLE *(Include Security Classification)*: See Box 19

12. PERSONAL AUTHOR(S): Steven D. Branch, B.B.A., Capt, USAF

13a. TYPE OF REPORT: MS Thesis

13b. TIME COVERED: FROM ______ TO ______

14. DATE OF REPORT *(Year, Month, Day)*: 1987 December

15. PAGE COUNT: 100

16. SUPPLEMENTARY NOTATION:

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
05	01	

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*:
Management Information Systems,
Information Systems,
Systems Analysis

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

Title: A Comparison of Information System Development Methodologies

Thesis Chairman: Kenneth R. Jennings, Captain, USAF
Assistant Professor of Management and
Organizational Behavior

Approved for public release: [illegible]
[illegible] E. [illegible]
7 Jan 88
Dean for [illegible]
Air Force [illegible] (AFC)
Wright-Patterson [illegible]

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT: ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL: Kenneth R. Jennings, Capt, USAF

22b. TELEPHONE *(Include Area Code)*: (513) 255-4529

22c. OFFICE SYMBOL: AFIT/LSR

DD Form 1473, JUN 86 *Previous editions are obsolete.*

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

This research critically evaluated a select sampling of current information systems development methodologies. The research had two primary objectives. The first was to enhance the manager's understanding of information systems and design methodologies. The second was to provide a tool to assist managers in deliberately choosing which methodology best fits their own specific needs.

Two sets of attributes were selected as the basis for an evaluative framework to compare methodologies. One set was chosen based on an information systems development life-cycle model. The intent was to specifically compare the utility of the methodologies for development throughout the complete range of information systems development. Another set of attributes was derived from factors that contribute to the institutionalization of the change represented by the information system in the organization. Each of the selected methodologies was then compared in relation to its degree of coverage of the attributes in the framework.

The research led to several conclusions. First, structured methodologies have tended toward a narrow focus in support of logical and physical design rather than expanding into a broader framework including proactive management of the change process. Second, socio-technical methodologies inherently paid more attention to factors which provide both a complete requirements analysis and support for the institutionalization of the change process. However, results from their use provide few specifics which can be easily translated into program code. Finally, since computer aided software engineering (CASE) methodologies appear most promising in their ability to allow development efforts to focus on the actual problem at hand vice the complex aspects of the solution to the problem, a merger of CASE with Socio-technical approaches is recommended.

END
FILMED
MARCH, 1988
DTIC